

## Ejercicios de autómatas y expresiones regulares

*A continuación se incluyen algunas partes de las especificaciones oficiales de varios lenguajes de programación. Construye los autómatas finitos y expresiones regulares de los elementos descritos.*

- **C Identifiers [Kernighan]**

An identifier is a sequence of letters and digits. The first character must be a letter; the underscore `_` is considered a letter. Uppercase and lowercase letters are considered different.

- **Java Integer Literals [Gosling]**

An integer literal may be expressed in decimal (base 10), hexadecimal (base 16), or octal (base 8). An integer literal is of type `long` if it is suffixed with an ASCII letter `L` or `l` (ell); otherwise it is of type `int`. The suffix `L` is preferred, because the letter `l` (ell) is often hard to distinguish from the digit `1` (one).

A decimal numeral is either the single ASCII character `0`, representing the integer zero, or consists of an ASCII digit from `1` to `9`, optionally followed by one or more ASCII digits from `0` to `9`, representing a positive integer.

A hexadecimal numeral consists of the leading ASCII characters `0x` or `0X` followed by one or more ASCII hexadecimal digits and can represent a positive, zero, or negative integer. Hexadecimal digits with values `10` through `15` are represented by the ASCII letters `a` through `f` or `A` through `F`, respectively; each letter used as a hexadecimal digit may be uppercase or lowercase.

An octal numeral consists of an ASCII digit `0` followed by one or more of the ASCII digits `0` through `7` and can represent a positive, zero, or negative integer.

Note that octal numerals always consist of two or more digits; `0` is always considered to be a decimal numeral-not that it matters much in practice, for the numerals `0`, `00`, and `0x0` all represent exactly the same integer value.

- **Scheme Integer Literals [Clinger]**

A number may be written in binary, octal, decimal, or hexadecimal by the use of a radix prefix. The radix prefixes are `#b` (binary), `#o` (octal), `#d` (decimal), and `#x` (hexadecimal). With no radix prefix, a number is assumed to be expressed in decimal.

- **Java Floating-Point Literals [Gosling]**

A floating-point literal has the following parts: a whole-number part, a decimal point (represented by an ASCII period character), a fractional part, an exponent, and a type suffix. The exponent, if present, is indicated by the ASCII letter `e` or `E` followed by an optionally signed integer.

At least one digit, in either the whole number or the fraction part, and either a decimal point, an exponent, or a float type suffix are required. All other parts are optional.

A floating-point literal is of type `float` if it is suffixed with an ASCII letter `F` or `f`; otherwise its type is `double` and it can optionally be suffixed with an ASCII letter `D` or `d`.

- **C Comments [Kernighan]**

The characters `/*` start a comment that ends with the characters `*/`. Comments cannot be nested nor can they be inside a string or character literal.